



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/692,765	10/19/2000	Thomas E. Saulpaugh	5181-65700	8734
58467	7590	04/02/2008	EXAMINER	
MHKKG/SUN P.O. BOX 398 AUSTIN, TX 78767			PATEL, ASHOKKUMAR B	
			ART UNIT	PAPER NUMBER
			2154	
			MAIL DATE	
			04/02/2008	DELIVERY MODE
				PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.



UNITED STATES PATENT AND TRADEMARK OFFICE

Commissioner for Patents
United States Patent and Trademark Office
P.O. Box 1450
Alexandria, VA 22313-1450
www.uspto.gov

**BEFORE THE BOARD OF PATENT APPEALS
AND INTERFERENCES**

Application Number: 09/692,765
Filing Date: October 19, 2000
Appellant(s): SAULPAUGH ET AL.

Robert C. Kowert
For Appellant

EXAMINER'S ANSWER

This is in response to the appeal brief filed 11/21/2005 and 01/07/2008 appealing from the Office action mailed 05/23/2005.

(1) Real Party in Interest

A statement identifying the real party in interest is contained in the brief.

(2) Related Appeals and Interferences

The examiner is not aware of any related appeals, interferences, or judicial proceedings which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

(3) Status of Claims

The statement of the status of claims contained in the brief is correct.

(4) Status of Amendments After Final

No amendment after final has been filed.

(5) Summary of Claimed Subject Matter

The summary of claimed subject matter contained in the brief is correct.

(6) Grounds of Rejection to be Reviewed on Appeal

The appellant's statement of the grounds of rejection to be reviewed on appeal is correct.

(7) Claims Appendix

The copy of the appealed claims contained in the Appendix to the brief is correct.

(8) Evidence Relied Upon

- a. Bass et al., US 6, 549, 956
- b. Meltzer et al., US 6, 542, 912

(9) Grounds of Rejection

The following ground(s) of rejection are applicable to the appealed claims:

Claims 1-11, 14-24, 27-33 and 36- 46 are rejected under 35 U.S.C. 102(e) as being anticipated by Bass et al. (hereinafter Bass) (US 6, 549, 956) and claims 12, 13, 25, 26, 34, 35, 47 and 48 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bass et al. (hereinafter Bass) (US 6, 549, 956) in view of Meltzer et al. (hereinafter Meltzer) (US 6, 542, 912).

* * * * *

Claim Rejections - 35 USC § 102

2. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

3. Claims 1-11, 14-24, 27-33 and 36- 46 are rejected under 35 U.S.C. 102(e) as being anticipated by Bass et al. (hereinafter Bass) (US 6, 549, 956)

Referring to claim 1,

The reference teaches a method for handling events in a distributed computing environment (col.1, line 63 through col.2, line 3 and col.3, lines 6-9, “The domains may be two separate corporations acting as business partners. Each domain represents a

separate and distinct network of computers within the corporations Intranet. The two domains communicate via the Internet 11."), comprising:

receiving a message in a data representation language (col.3, lines 43-50) sent to a client platform in the distributed computing environment from a service in the distributed computing environment, wherein the message includes a data representation language representation of an event generated by the service (col.2, lines 4-9, lines 15-31, Note: The reference teaches" For transport across the network, the inventive channel adapters convert the event information into a format acceptable by the network. The delivered information is then reconverted back into the event format for use in the other domain." Thereby the reference discloses that an event (message) can be represented in any data representation language and will be converted back into the event format for use in the other domain (a client platform)); and

sending the data representation language representation of the event to one or more processes registered to receive the event from the service (col.2, lines 9-15, "Likewise, a process adapter can subscribe to an event type from a channel adapter that is listed on an event type list from its peer channel adapter. When an event is received via the channel adapter and re-published into the domain, the subscribing process adapter will receive the event.")

Referring to claim 2,

The reference teaches the method as recited in claim 1, further comprising;

receiving a data representation language schema on the client platform, wherein said data representation language schema defines a message interface for a set of events generated by the service; and

generating an event message endpoint for the client platform according to the data representation language schema, wherein said receiving a message and said sending the data representation language representation of the event to one or more processes are performed by the event message endpoint.(col.3, lines 43-50, col.4, lines 43 through col.5, line 15, col.2, lines 4-15, "Each channel adapter is initialized with a set of events it will export to its peer at the other domain. The two channel adapters handshake with these sets of events. Process adapters within each domain can then publish events of the exported type and expect the event to be carried to the other domain via the network protocol. Likewise, a process adapter can subscribe to an event type from a channel adapter that is listed on an event type list from its peer channel adapter. When an event is received via the channel adapter and re-published into the domain, the subscribing process adapter will receive the event." Thereby the reference teaches that each channel adapter is initialized with a set of events it will export to its peer at the other domain and they handshake with sets of events even though they are in a completely different domains (receiving a data representation language schema on the client platform, wherein said data representation language schema defines a message interface for a set of events generated by the service). Thereby the reference also teaches" Process adapters within each domain can then publish events of the exported type and expect the event to be carried to the other

domain via the network protocol. Likewise, a process adapter can subscribe to an event type from a channel adapter that is listed on an event type list from its peer channel adapter. When an event is received via the channel adapter and re-published into the domain, the subscribing process adapter will receive the event" (generating an event message endpoint for the client platform according to the data representation language schema, wherein said receiving a message and said sending the data representation language representation of the event to one or more processes are performed by the event message endpoint.)

Referring to claim 3,

The reference teaches the method ms recited in claim 2, further comprising the event message endpoint subscribing to one or more of the set of events generated by the service, wherein the service is configured to send messages including data representation language representations of an event to subscribers to the event when the event is generated. (col.3, lines 43-50).

Referring to claim 4,

The reference teaches the method as recited in claim 2, wherein the data representation language message from the service includes an authentication credential for the service, the method further comprising the event message endpoint using the authentication credential for the service to authenticate the data representation language message as being from the service. (col.4, line57 through col.5, line15).

Referring to claim 5,

The reference teaches the method as recited in claim 2, further comprising the event message endpoint verifying type correctness of the data representation language message according to the data representation language schema subsequent to said receiving a message. (col.2, lines 24-27, col.3, lines 45-50).

Referring to claim 6,

The reference teaches the method as recited in claim 2, wherein the data representation language schema defines a set of messages that the service may send to the event message endpoint, the method further comprising the event message endpoint verifying the correctness of the data representation language message from the service according to the data representation language schema. (col.2, lines 24-27, col.3, lines 45-50).

Referring to claim 7,

The reference teaches the method as recited in claim 2, further comprising each of the one or more processes registering interest in one or more of the set of events generated by the service with the event message endpoint subsequent to said generating an event message endpoint. (Fig.1, col. 4, line 43 through col.5, line15).

Referring to claim 8,

The reference teaches the method as recited in claim 7, wherein said registering interest in one or more of the set of events comprises each of the one or more processes providing an event handler callback method to the event message endpoint; wherein said sending the data representation language representation of the event to one or more processes registered to receive the event from the service(col.4, line 57-

60, Note: Thereby the reference teaches that the processes as well as the adapters are configured to do the claimed element.) comprises:

the event message endpoint calling an event handler method of each process registered with the event message endpoint to the event; and the event message endpoint passing the data representation language representation of the event to each called event handler (col.3, lines 22-50, Note: The cannel adapters are capable of executing the task as claimed.)

Referring to claim 9,

The reference teaches the method ms recited in claim 7, further comprising:

a process unregistering interest in a first event of the service; and the event message gate unsubscribing to the first event with the service subsequent to said unregistering; wherein the service is further configured to not send messages including data representation language representations of the first event to event message endpoints that are unsubscribed to the first event. (col.4, line57 through col.5, line 15).

Referring to claims 10 and 11,

The reference teaches the method as recited in claim 2, further comprising receiving the data representation language schema of the service in a service advertisement of the service.(col.3, lines 43-50, col.4, lines 43 through col.5, line 15, col.2, lines 4-15, “Each channel adapter is initialized with a set of events it will export to its peer at the other domain. The two channel adapters handshake with these sets of events. Process adapters within each domain can then publish events of the exported type and expect the event to be carried to the other domain via the network protocol. Likewise, a

process adapter can subscribe to an event type from a channel adapter that is listed on an event type list from its peer channel adapter. When an event is received via the channel adapter and re-published into the domain, the subscribing process adapter will receive the event.", and wherein the one or more processes are executing within the client platform (Fig.1, element 18 and 19).

Referring to claim 14,

Claim 14 is a claim to the device configured for carrying out the steps of method of claim 1. Therefore, claim 14 is rejected for the reasons set forth for claim 1.

Referring to claim 15,

Claim 15 is a claim to the device configured for carrying out the steps of method of claim 2. Therefore, claim 15 is rejected for the reasons set forth for claim 2.

Referring to claim 16,

Claim 16 is a claim to the device configured for carrying out the steps of method of claim 5. Therefore, claim 16 is rejected for the reasons set forth for claim 5.

Referring to claim 17,

Claim 17 is a claim to the device configured for carrying out the steps of method of claim 6. Therefore, claim 17 is rejected for the reasons set forth for claim 6.

Referring to claim 18,

Claim 18 is a claim to the device configured for carrying out the steps of method of claim 10. Therefore, claim 18 is rejected for the reasons set forth for claim 10.

Referring to claim 19,

Claim 19 is a claim to the device configured for carrying out the steps of method of claim 3. Therefore, claim 19 is rejected for the reasons set forth for claim 3.

Referring to claim 20,

Claim 20 is a claim to the device configured for carrying out the steps of method of claim 4. Therefore, claim 20 is rejected for the reasons set forth for claim 4.

Referring to claim 21,

Claim 21 is a claim to the device configured for carrying out the steps of method of claim 7. Therefore, claim 21 is rejected for the reasons set forth for claim 7.

Referring to claim 22,

Claim 22 is a claim to the device configured for carrying out the steps of method of claim 8. Therefore, claim 22 is rejected for the reasons set forth for claim 8.

Referring to claim 23,

Claim 23 is a claim to the device configured for carrying out the steps of method of claim 9. Therefore, claim 23 is rejected for the reasons set forth for claim 9.

Referring to claim 24,

Claim 24 is a claim to the device configured for carrying out the steps of method of claim 11. Therefore, claim 24 is rejected for the reasons set forth for claim 11.

Referring to claim 27,

The reference teaches a device, comprising:
a processor; a memory coupled to said processor (Fig.1-3);
a service process Configured to:
generate an event;

generate a message in a data representation language, wherein the message includes a data representation language representation of the event generated by the service process(col.2, lines 4-9, lines 15-31, Note: The reference teaches "For transport across the network, the inventive channel adapters convert the event information into a format acceptable by the network. The delivered information is then reconverted back into the event format for use in the other domain." Thereby the reference discloses that an event (message) can be represented in any data representation language and will be converted back into the event format for use in the other domain (a client platform)); and send the message to one or more event message gate units (col.2, lines 9-15, "Likewise, a process adapter can subscribe to an event type from a channel adapter that is listed on an event type list from its peer channel adapter. When an event is received via the channel adapter and re-published into the domain, the subscribing process adapter will receive the event.") in the distributed computing environment(col.1, line 63 through col.2, line 3 and col.3, lines 6-9, "The domains may be two separate corporations acting as business partners. Each domain represents a separate and distinct network of computers within the corporations Intranet. The two domains communicate via the Internet 11.");

wherein each of the one or more event message gate units are operable to distribute the data representation language representation of the event sent in the message from the service process to one or more processes registered to receive the event from the service process (Fig.1, col.3, lines 5-50).

Referring to claim 28,

The reference teaches the device as recited in claim 27, wherein the device further comprises a service message gate unit, wherein said generating a message and said sending the message are performed by the service message gate unit on behalf of the service process (col.col.3, lines 22-25).

Referring to claim 29,

Claim 29 is a claim to the device configured for carrying out the steps of method of claim 2. Therefore, claim 15 is rejected for the reasons set forth for claim 2.

Referring to claim 30,

Claim 30 is a claim to the device configured for carrying out the steps of method of claim 6. Therefore, claim 30 is rejected for the reasons set forth for claim 6.

Referring to claim 31,

Claim 31 is a claim to the device configured for carrying out the steps of method of claim 10. Therefore, claim 31 is rejected for the reasons set forth for claim 10.

Referring to claim 32,

Claim 32 is a claim to the device configured for carrying out the steps of method of claim 3. Therefore, claim 32 is rejected for the reasons set forth for claim 3.

Referring to claim 33,

Claim 33 is a claim to the device configured for carrying out the steps of method of claim 4. Therefore, claim 33 is rejected for the reasons set forth for claim 4.

Referring to claim 36,

Claim 36 is a claim to a computer readable carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement the steps of method of claim 1. Therefore, Claim 36 is rejected for the reasons set forth for claim 1.

Referring to claim 37,

Claim 37 is a claim to a computer readable carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement the steps of method of claim 2. Therefore, Claim 37 is rejected for the reasons set forth for claim 2.

Referring to claim 38,

Claim 38 is a claim to a computer readable carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement the steps of method of claim 3. Therefore, Claim 38 is rejected for the reasons set forth for claim 3.

Referring to claim 39,

Claim 39 is a claim to a computer readable carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement the steps of method of claim 4. Therefore, Claim 39 is rejected for the reasons set forth for claim 4.

Referring to claim 40,

Claim 40 is a claim to a computer readable carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement

the steps of method of claim 5. Therefore, Claim 40 is rejected for the reasons set forth for claim 5.

Referring to claim 41,

Claim 41 is a claim to a computer readable carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement the steps of method of claim 6. Therefore, Claim 41 is rejected for the reasons set forth for claim 6.

Referring to claim 42,

Claim 42 is a claim to a computer readable carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement the steps of method of claim 7. Therefore, Claim 42 is rejected for the reasons set forth for claim 7.

Referring to claim 43,

Claim 43 is a claim to a computer readable carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement the steps of method of claim 8. Therefore, Claim 43 is rejected for the reasons set forth for claim 8.

Referring to claim 44,

Claim 44 is a claim to a computer readable carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement the steps of method of claim 9. Therefore, Claim 44 is rejected for the reasons set forth for claim 9.

Referring to claim 45,

Claim 45 is a claim to a computer readable carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement the steps of method of claim 10. Therefore, Claim 45 is rejected for the reasons set forth for claim 10.

Referring to claim 46,

Claim 46 is a claim to a computer readable carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement the steps of method of claim 11. Therefore, Claim 46 is rejected for the reasons set forth for claim 11.

Claim Rejections - 35 USC § 103

4. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negated by the manner in which the invention was made.

5. Claims 12, 13, 25, 26, 34, 35, 47 and 48 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bass et al. (hereinafter Bass) (US 6, 549, 956) in view of Meltzer et al. (hereinafter Meltzer) (US 6, 542, 912)

Referring to claims 12 and 13,

Keeping in mind the teachings of the reference Bass as stated above, the reference explicitly fails to teach wherein the event is a Java event and the data representation

language is extensible Markup Language (XML). The reference Meltzer teaches “By translating the elements of the XML document into JAVA events or other programming structures that are suitable for use by the transaction processing front end of the respective nodes enables rich functionality at nodes utilizing the documents being traded.” (col.14, lines 25-32). Thereby not only the reference teaches the claimed elements but provides the motivation to do so along with. Therefore, it would have been obvious to one having ordinary skill in the art at the time of invention was made to incorporate the teachings of Meltzer into the teachings of Bass such that the transaction process front end (channel adapters are configured to) is able to operate in a publish and subscribe architecture that enables the addition of new listener programs without the knowledge of or impact on other listening programs in the system as taught by the reference.

Referring to claims 25, 26, 34 and 35,

Claims 25, 26, 34 and 35 are claims to the device configured for carrying out the steps of method of claims 12 and 13. Therefore, claims 25, 26, 34 and 35 are rejected for the reasons set forth for claims 12 and 13.

Referring to claims 47 and 48,

Claims 47 and 48 are claims to a computer readable carrier medium comprising program instructions, wherein the program instructions are computer-executable to implement the steps of method of claims 12 and 13. Therefore, Claims 47 and 48 are rejected for the reasons set forth for claims 12 and 13.

* * * * *

(10) Response to Argument

First Ground of Rejection:

Claims 1, 11, 14, 24, 36 and 46:

Appellant's argument:

“However, Examiner reads too much into the actual teachings of Bass. The Examiner is arguing that the phrase “a format acceptable by the network” discloses the use of any data representation language. However, a data representation language is a particular type of language. As is well understood by anyone of ordinary skill in the art, a data representation language (such as XML) has a particular structure as a language and is a language for representing (or describing) data or content. There is clearly no teaching in Bass that any messages are sent in a particular data representation language. Nor is there any teaching in Bass that the events are represented in such a language. Without some clear teaching by Bass regarding the use of a data representation language, Bass cannot be said to anticipate a message in a data representation language including a data representation language representation of an event.”

“Apparently the Examiner is arguing that SMTP is a data representation language. However, SMTP is a protocol, not a data representation language.”

“The Examiner's reference to the use of SMTP supports Appellants' argument. As is well understood by any one of ordinary skill in the art, the Simple Mail Transfer

Protocol does not require any particular language, let alone a data representation language.”

“And Bass clearly does not mention using messages in a data representation language.”

“Bass fails to mention anything about XML, as admitted by the Examiner regarding the rejection of claim 13, discussed below. As noted above, Bass fails to mention any data representation language. The prior art does not teach the use of a data representation language to represent events in messages between entities in a distributed computing environment.”

“Additionally, Bass fails to anticipate that the message includes a data representation language representation of an event generated by the service.”

“Furthermore, Bass fails to anticipate sending the data representation language representation of the event to one or more processes registered to receive the event from the service.”

“As noted above, Bass teaches only translating event information into a format suitable for transmission over the Internet via any of a number exiting network protocols (such as TCPX, SMTP, FTP, etc). However, nowhere does Bass mention that the event information is a data representation language representation of an event.”

“Arguments similar to those presented above regarding claim 1 apply to claims 14 and 36 as well.”

Examiner’s response:

Examiner would like to reveal Bass’s teachings and present in steps:

Step 1:

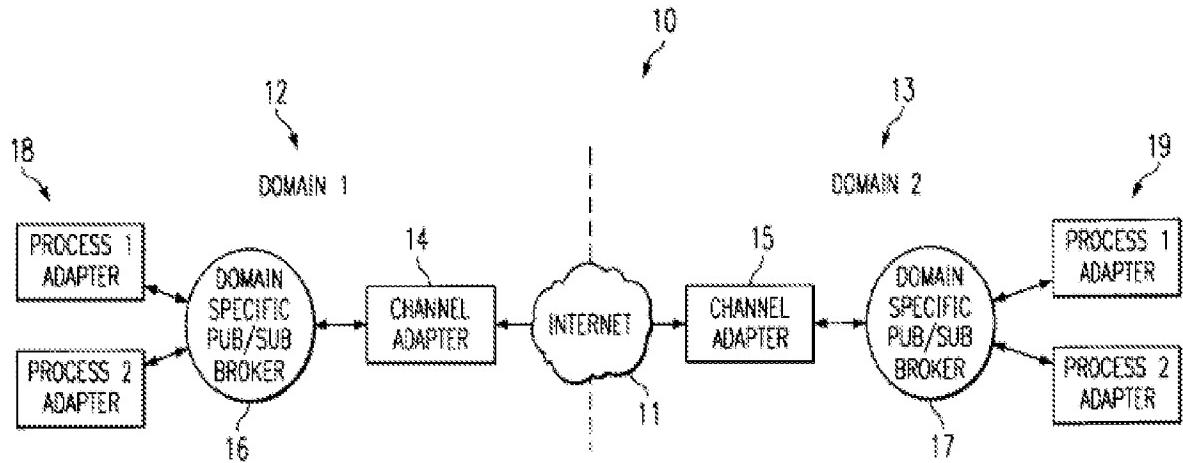


FIG. 1

Bass's environment is a set up for it's teaching's application at col. 3, line 5-14, "FIG. 1 depicts a high level view 10 of two domains, domain 112 and domain 213. The domains may be two separate corporations acting as business partners. Each domain represents a separate and distinct network of computers within the corporations Intranet. The two domains communicate via the Internet 11. Each domain includes an inventive channel adapter 14, 15, and a domain specific publication/subscription (PUB/SUB) service 16, 17. PUB/SUB service is the message broker service used within the domain."

Thus, communication between distinct networks, (distributed computing environment), is via Internet is clearly understood.

Step 2:

Bass teaches at col. 3, line 22-42, "The inventive channel adapters 14 and 15 permit dissimilar systems to communicate with each other. The adapters acts as a proxy or an agent for communications and data exchanges between the business applications. Each adapter interfaces with its respective broker and the Internet. Note that although the Internet is depicted and described herein, other networks, such as a WAN or LAN, may be used in place of the Internet. Thus, the broker views the adapter as another process adapter within the domain to which events are published or subscribed. Furthermore, the channel adapter is configured by the domain administrator in the same manner as a process adapter in that the adapter is represented on an administrator screen as an icon which can be set to specify which events couple which processes (including the adapter). The adapter also converts communications and data into a format for transmission on the Internet (or other network). The format may use transmission control protocol/Internet protocol (TCP/IP), simple mail transport protocol (SMTP), File Transfer Protocol (FTP), or whatever protocol is useable by the connecting network."

Thus, SMTP is used for transmission of formatted data is clearly understood.

Step 3:

Bass continues teaching at col. 3, line 42-50, "Thus, the flow of information for FIG. 1 is as follows. An event originating in a process adapter 18 in domain 112 is delivered to the subscribing channel adapter 14. This event would be transformed into (for example) an e-mail via SMTP, and mailed from the channel adapter through the Internet to domain 2 13. The e-mail is received by the channel adapter

15 and re-transformed back into the event. The channel adapter then delivers the event to any subscribing process adapters within the domain.”

Thus, **SMTP is used** for transmission of formatted **data which is “event” transformed into an e-mail** is clearly understood.

Further, Bass teaches that the E-mail containing data which is an event is delivered to a specific E-mail Address, as it indicates in col. 4, line 3.

Also in col. 7, line 41-45 in reference to Fig.3, Bass also teaches “Another thread is the outbound transmitting thread 305 that converts events within the outbound queue into Internet format messages, and then sends these messages to an SMTP server (or other server types) for transmission over the Internet (or other network types).”

Thus, the event converted into email contains the event data and e-mail address into the internet format messages. Thus, the event email transmitted over the internet is in a data representation language format (internet format displays “emails” on the computer screen.)

Thus, “ a message in a data representation language” is the email, which is “the data representation language representation of the event”, taught by Bass.

And thus, “ a message in a data representation language” is sent, which is “the data representation language representation of the event” is delivered to any subscribing process adapters.

Step 4:

Claim 1 recites: A method for handling events in a distributed computing environment (col. 3, line 5-14, as explained in Step 1), comprising:

"receiving a message in a data representation language ", as such, the message consists of a data representation language (as explained in Step 2 and Step 3), further, claim 1 goes on reciting "the message includes a data representation language representation of an event", as such, the message includes and represents an event (as explained in Step 2 and Step 3), and

further, claim 1 goes on reciting "sending the data representation language representation of the event to one or more processes", as such, the message representing and including the event is sent to one or more processes (as explained in Step 2 and Step 3).

Therefore, Bass does mention using messages in a data representation language, and additionally, Bass anticipates that the message includes a data representation language representation of an event generated by the service.

Furthermore, Bass also anticipates sending the data representation language representation of the event to one or more processes registered to receive the event from the service.

Bass's failure "to mention anything about XML, as admitted by the Examiner regarding the rejection of claim 13, discussed below", **has nothing to do with** claims 1, 11, 14, 24, 36 and 46, since any of these claim's language makes no indication of XML.

It is the claims that define the claimed invention, and it is claims, not specifications that are anticipated or unpatentable. *Constant v. Advanced Micro-Devices Inc.*, 7 USPQ2d 1064.

Response presented above regarding claim 1 applies to claims 14 and 36 as well.

Claims 2, 7, 15, 21, 37 and 42:

Appellant's argument:

"The Examiner has not cited any portion of Bass that teaches a data representation language schema defining a message interface for a set of events. Instead, Basses Bass teaches that each channel adapter includes two different interfaces for communicating event information."

"Bass does not teach that his event export lists make up a *data representation language schema*."

Thus, the exchange of exported event lists cited by the Examiner does not teach anything regarding receiving a data representation language schema defining a message interface.

Additionally, Bass does not teach generating an event message endpoint for the client platform according to the data representation language schema."

"Thus, Bass clearly fails to disclose generating an event message endpoint for the client platform according to the data representation language schema."

"Similar arguments as those presented above apply to claims 15 and 37 as well."

Examiner's response:

Bass teaches at col. 3, line 42-50, "Thus, the flow of information for FIG. 1 is as follows. An event originating in a process adapter 18 in domain 1 12 is delivered to the subscribing channel adapter 14. This event would be transformed into (for example) an e-mail via SMTP, and mailed from the channel adapter through the Internet to domain 2 13. The e-mail is received by the channel adapter 15 and re-transformed back into the event. The channel adapter then delivers the event to any subscribing process adapters within the domain."

Thus, "channel adapters are end points for receiving as well as delivering events between two domains" is understood.

Bass teaches at col. 3, line 59-62, "Each channel adapter includes a protocol interface 22, 23. This interface comprises the network specific protocols that enable the adapter to couple with the Internet 11. The protocol interface facilitates the conversion of the events into a network transportable format, e.g. e-mail, and vice versa. Each channel adapter also includes a configuration interface 28."

Thus, the channel adapters include interfaces protocol interfaces facilitating the conversion of the events into a network transportable format, e.g. e-mail, and vice versa (wherein data representation language schema defines a message interface) is understood.

Now, Bass goes on teaching at col.2, lines 4-15, "Each channel adapter is initialized with a set of events it will export to its peer at the other domain. The two channel adapters handshake with these sets of events. Process adapters within each domain can then publish events of the exported type and expect the event to be carried

to the other domain via the network protocol. Likewise, a process adapter can subscribe to an event type from a channel adapter that is listed on an event type list from its peer channel adapter. When an event is received via the channel adapter and re-published into the domain, the subscribing process adapter will receive the event."(generating an event message endpoint for the client platform according to the data representation language schema, wherein said receiving a message and said sending the data representation language representation of the event to one or more processes are performed by the event message endpoint.)

Thereby the reference teaches that each channel adapter is initialized with a set of events it will export to its peer at the other domain and they handshake with sets of events even though they are in a completely different domains (receiving a data representation language schema on the client platform, wherein said data representation language schema defines a message interface for a set of events generated by the service).

Response presented above applies to claims 15 and 37 as well.

Claims 3, 19 and 38:

Appellant's argument:

"Regarding claim 3, Bass fails to disclose the event message endpoint subscribing to one or more of the set of events generated by the service, wherein the service is configured to send messages including data representation language representations of an event to subscribers to the event when the event is generated."

"However, as discussed above regarding claims 1 and 2, Bass fails to teach anything regarding a service configured to send messages including *data representation language representations of events.*"

"Bass does not describe these protocol messages as being messages in a data representation language. Furthermore, nowhere does Bass mention a service configured to send messages including data representation language representations of an event."

"Similar arguments as those presented above apply to claims 19 and 38 as well."

Examiner's response:

Bass goes on teaching at col.2, lines 4-15, "Each channel adapter is initialized with a set of events it will export to its peer at the other domain. The two channel adapters handshake with these sets of events. Process adapters within each domain can then publish events of the exported type and expect the event to be carried to the other domain via the network protocol. Likewise, a process adapter can subscribe to an event type from a channel adapter that is listed on an event type list from its peer channel adapter. When an event is received via the channel adapter and re-published into the domain, the subscribing process adapter will receive the event."

Thus, "channel adapters are end points for receiving as well as delivering events between two domains" is clearly understood.

And "a process adapter can subscribe to an event type from a channel adapter that is listed on an event type list from its peer channel adapter" is clearly understood.

And, thereby the event message endpoint subscribing to one or more of the set of events generated by the service, wherein the service is configured to send messages including data representation language representations of an event, as explained in claim 1, to subscribers to the event when the event is generated.” is taught by Bass.

Response presented above applies to claims 19 and 38 as well.

Claims 4, 20 and 39:

Appellant's argument:

“In regards to claim 4, Bass fails to disclose wherein the data representation language message from the service includes an authentication credential for the service. Bass additionally fails to disclose using the authentication credential for the service to authenticate the data representation language message as being from the service.

“Similar arguments as those presented above apply to claims 20 and 39 as well.”

Examiner's response:

Bass teaches at col. 3, line 64 through col. 4, line 3, “Each channel adapter also includes a configuration interface 28. This interface allows the channel adapter to be configured by an administrator. The administrator (not shown) could select the protocols used for the interfaces and specify what events or types of events would flow through the channel adapter and configures protocol specific parameters, e.g. e-mail addresses.”

Thus, “service includes an authentication credential for the service.” is clearly understood.

Response presented above applies to claims 20 and 39 as well.

Claims 5, 16 and 40:

Appellant's argument:

"Regarding claim 5, Bass fails to disclose the event message endpoint verifying type correctness of the data representation language message according to the data representation language schema."

"In fact, nowhere does Bass make any reference whatsoever to verifying type correctness of a data representation language message according to a data representation language schema."

"Similar arguments as those presented above apply to claims 16 and 40 as well.."

Examiner's response:

As indicated above for claim 2, Bass teaches at col. 3, line 42-50, "Thus, the flow of information for FIG. 1 is as follows. An event originating in a process adapter 18 in domain 1 12 is delivered to the subscribing channel adapter 14. This event would be transformed into (for example) an e-mail via SMTP, and mailed from the channel adapter through the Internet to domain 2 13. The e-mail is received by the channel adapter 15 and re-transformed back into the event. The channel adapter then delivers the event to any subscribing process adapters within the domain."

Thus, "channel adapters are end points for receiving as well as delivering events between two domains" is clearly understood.

Bass teaches at col. 3, line 59-62, "Each channel adapter includes a protocol interface 22, 23. This interface comprises the network specific protocols that enable the adapter to couple with the Internet 11. The protocol interface facilitates the conversion of the events into a network transportable format, e.g. e-mail, and vice versa. Each channel adapter also includes a configuration interface 28."

Thus, the channel adapters include interfaces protocol interfaces facilitating the conversion of the events into a network transportable format, e.g. e-mail, and vice versa (wherein data representation language schema defines a message interface) is clearly understood.

Now, Bass goes on teaching at col.2, lines 4-15, "Each channel adapter is initialized with a set of events it will export to its peer at the other domain. The two channel adapters handshake with these sets of events. Process adapters within each domain can then publish events of the exported type and expect the event to be carried to the other domain via the network protocol. Likewise, a process adapter can subscribe to an event type from a channel adapter that is listed on an event type list from its peer channel adapter. When an event is received via the channel adapter and re-published into the domain, the subscribing process adapter will receive the event."(generating an event message endpoint for the client platform according to the data representation language schema, wherein said receiving a message and said sending the data representation language representation of the event to one or more processes are performed by the event message endpoint.)

Thereby the reference teaches that each channel adapter is initialized with a set of events it will export to its peer at the other domain and they handshake with sets of events even though they are in a completely different domains (receiving a data representation language schema on the client platform, wherein said data representation language schema defines a message interface for a set of events generated by the service).

Now that Bass has taught that the “data representation language schema defines a message interface for a set of events generated by the service”, Bass further teaches at col. 4, line 4-17, “Each channel adapter also includes a guaranteed message delivery mechanism 24, 25. This mechanism receives, stores and transmits events, as well as monitors and tracks the delivery states of the events. The delivery state of an event transitions as the two channel adapters process the event. The state of the event would start with an outbound send state on the sending channel adapter of the network, and would end with both an outbound acknowledge state on the sending channel adapter and an inbound acknowledge state on the receiving channel adapter. At the point in time when the state is inbound acknowledged, then the sending channel adapter knows that the event has been published within the PUB/SUB domain on the receiving side of the network.”

Thus, verifying type correctness of the data representation language message according to the data representation language schema subsequent to receiving a message is clearly understood.

Response presented above applies to claims 16 and 40 as well.

Claims 6, 17 and 41:

Appellant's argument:

Regarding claim 6, Bass fails to anticipate wherein the data representation language schema defines a set of messages that the service may send to the event message endpoint and further fails to teach the event message endpoint verifying the correctness of the data representation language message from the service according to the data representation language schema."

"Similar arguments as those presented above apply to claims 17 and 41 as well."

Examiner's response:

As indicated above for claim 2, Bass teaches at col. 3, line 42-50, "Thus, the flow of information for FIG. 1 is as follows. An event originating in a process adapter 18 in domain 1 12 is delivered to the subscribing channel adapter 14. This event would be transformed into (for example) an e-mail via SMTP, and mailed from the channel adapter through the Internet to domain 2 13. The e-mail is received by the channel adapter 15 and re-transformed back into the event. The channel adapter then delivers the event to any subscribing process adapters within the domain."

Thus, "channel adapters are end points for receiving as well as delivering events between two domains" is clearly understood.

Bass teaches at col. 3, line 59-62, "Each channel adapter includes a protocol interface 22, 23. This interface comprises the network specific protocols that enable the adapter to couple with the Internet 11. The protocol interface facilitates the conversion

of the events into a network transportable format, e.g. e-mail, and vice versa. Each channel adapter also includes a configuration interface 28."

Thus, the channel adapters include interfaces protocol interfaces facilitating the conversion of the events into a network transportable format, e.g. e-mail, and vice versa (wherein data representation language schema defines a message interface) is clearly understood.

Now, Bass goes on teaching at col.2, lines 4-15, "Each channel adapter is initialized with a set of events it will export to its peer at the other domain. The two channel adapters handshake with these sets of events. Process adapters within each domain can then publish events of the exported type and expect the event to be carried to the other domain via the network protocol. Likewise, a process adapter can subscribe to an event type from a channel adapter that is listed on an event type list from its peer channel adapter. When an event is received via the channel adapter and re-published into the domain, the subscribing process adapter will receive the event."(generating an event message endpoint for the client platform according to the data representation language schema, wherein said receiving a message and said sending the data representation language representation of the event to one or more processes are performed by the event message endpoint.)

Thereby the reference teaches that each channel adapter is initialized with a set of events it will export to its peer at the other domain and they handshake with sets of events even though they are in a completely different domains (receiving a data representation language schema on the client platform, wherein said data

representation language schema defines a message interface for a set of events generated by the service).

Now that Bass has taught that the “data representation language schema defines a message interface for a set of events generated by the service”, Bass further teaches at col. 4, line 4-17, “Each channel adapter also includes a guaranteed message delivery mechanism 24, 25. This mechanism receives, stores and transmits events, as well as monitors and tracks the delivery states of the events. The delivery state of an event transitions as the two channel adapters process the event. The state of the event would start with an outbound send state on the sending channel adapter of the network, and would end with both an outbound acknowledge state on the sending channel adapter and an inbound acknowledge state on the receiving channel adapter. At the point in time when the state is inbound acknowledged, then the sending channel adapter knows that the event has been published within the PUB/SUB domain on the receiving side of the network.”

Thus, verifying type correctness of the data representation language message according to the data representation language schema subsequent to receiving a message is clearly understood.

Response presented above applies to claims 17 and 41 as well.

Claims 8, 22 and 43:

Appellant’s argument:

“Regarding claim 8, Bass fails to disclose each of the one or more processes providing an event handler callback method to the event message endpoint.”

"Nowhere does Bass teach providing an event handler callback method to an event message endpoint."

"Bass further fails to teach the event message endpoint callback an event handler method of each process registered with the event message endpoint and the event message endpoint passing the data representation language representation of the event to each called event handler."

"Similar arguments as those presented above apply to claims 22 and 43 as well."

Examiner's response:

Bass teaches in col. 4, lines 43 through col. 5, line 6, Note that the terms publish and subscribe should not be used in describing the flow of control in sending an event across the system. For example, assume process adapter 1 19 of domain 213 has an event that is to be shared with any interested process adapters 18 of domain 1 12. The sending channel adapter 15 will receive the event from the process adapter 19 within its domain by subscribing to the event with the broker 17. Thus, when the event is published by the originating process adapter 19, the sending channel adapter 15 will receive the event, reformat the event and send it to the channel adapter 14 of domain 1 12 via Internet 11. The receiving channel adapter 14 reformats the event, and then publishes the event to the broker 16, which republishes the event to subscribing process adapter 18 within domain 112. Thus, prior to transfer of events between the domains, the respective Process and channel adapters of the domains must be configured to send and receive the different events."

Thus, “the event message endpoint callback an event handler method of each process registered with the event message endpoint and the event message endpoint passing the data representation language representation of the event to each called event handler.” Is understood.

Response presented above applies to claims 22 and 43 as well.

Claims 9, 23 and 44:

Appellant’s argument:

“Regarding claim 9, Bass fails to disclose, contrary to the Examiner’s assertion, a process unregistering interest in a first event of the service.”

“Additionally, Bass fails to the event message gate unsubscribing to the event with the service subsequent to the unregistering.”

“Similar arguments as those presented above apply to claims 23 and 44 as well.”

Examiner’s response:

Bass teaches at col.4, line 57 through col.5, line 15, “Thus, prior to transfer of events between the domains, the respective process and channel adapters of the domains must be configured to send and receive the different events. The mechanism for configuring the adapters is as follows. The configuration begins when an adapter signals the administrator that it has a new event to publish to interested processes. For example, assume process adapter 119 of domain 213 is publishing the new event. The administrator would configure the broker 17 to list channel adapter 15 as a subscriber to the event. Thus, adapter 15 would receive the event from the process, when the process adapter publishes the event. The administrator, via configuration interface 29,

would configure the channel adapter to send the event over the Internet to channel adapter 14 of domain 1. The channel adapter 15 would signal the channel adapter 14 that a new event is going to be received by channel adapter 14, and sends the updated export list to the other channel adapter. The channel adapter of domain 2 becomes capable of exporting the event. The administrator of domain 1 would configure the channel adapter 14 to publish the event to interested process adapter in domain 1. Note that in a PUB/SUB system, the publishers have no knowledge of the subscribers, and the subscribers have no knowledge of the publishers, as the brokers route published events to interested subscribers. Further note that each administrator is connected to each adapter within the domain managed by the administrator.

Thus, the event message gate unsubscribing as well as subscribing to the event with the service subsequent to the unregistering" is taught by Bass.

Response presented above applies to claims 23 and 44 as well.

Claims 10, 18 and 45:

Appellant's argument:

"Regarding claim 10, Bass does not disclose receiving the data representation language schema of the service in a service advertisement of the service."

"Similar arguments as those presented above apply to claims 18 and 45 as well."

Examiner's response:

As indicated above for claim 2, Bass teaches at col. 3, line 42-50, "Thus, the flow of information for FIG. 1 is as follows. An event originating in a process adapter 18 in domain 1 12 is delivered to the subscribing channel adapter 14. This event would be

transformed into (for example) an e-mail via SMTP, and mailed from the channel adapter through the Internet to domain 2 13. The e-mail is received by the channel adapter 15 and re-transformed back into the event. The channel adapter then delivers the event to any subscribing process adapters within the domain.”

Thus, “channel adapters are end points for receiving as well as delivering events between two domains” is understood.

Bass teaches at col. 3, line 59-62, “Each channel adapter includes a protocol interface 22, 23. This interface comprises the network specific protocols that enable the adapter to couple with the Internet 11. The protocol interface facilitates the conversion of the events into a network transportable format, e.g. e-mail, and vice versa. Each channel adapter also includes a configuration interface 28.”

Thus, the channel adapters include interfaces protocol interfaces facilitating the conversion of the events into a network transportable format, e.g. e-mail, and vice versa (wherein data representation language schema defines a message interface) is understood.

Now, Bass goes on teaching at col.2, lines 4-15, "Each channel adapter is initialized with a set of events it will export to its peer at the other domain. The two channel adapters handshake with these sets of events. Process adapters within each domain can then publish events of the exported type and expect the event to be carried to the other domain via the network protocol. Likewise, a process adapter can subscribe to an event type from a channel adapter that is listed on an event type list from its peer channel adapter. When an event is received via the channel adapter and re-published

into the domain, the subscribing process adapter will receive the event."(generating an event message endpoint for the client platform according to the data representation language schema, wherein said receiving a message and said sending the data representation language representation of the event to one or more processes are performed by the event message endpoint.)

Thereby the reference teaches that each channel adapter is initialized with a set of events it will export to its peer at the other domain and they handshake with sets of events even though they are in a completely different domains (receiving a data representation language schema on the client platform, wherein said data representation language schema defines a message interface for a set of events generated by the service).

Now that Now that Bass has taught that the "data representation language schema defines a message interface for a set of events generated by the service" , Bass teaches at col.4, line 57 through col.5, line 15, "Thus, prior to transfer of events between the domains, the respective process and channel adapters of the domains must be configured to send and receive the different events. The mechanism for configuring the adapters is as follows. The configuration begins when an adapter signals the administrator that it has a new event to publish to interested processes. For example, assume process adapter 119 of domain 213 is publishing the new event. The administrator would configure the broker 17 to list channel adapter 15 as a subscriber to the event. Thus, adapter 15 would receive the event from the process, when the process adapter publishes the event. The administrator, via configuration interface 29,

would configure the channel adapter to send the event over the Internet to channel adapter 14 of domain 1. The channel adapter 15 would signal the channel adapter 14 that a new event is going to be received by channel adapter 14, and sends the updated export list to the other channel adapter. The channel adapter of domain 2 becomes capable of exporting the event. The administrator of domain 1 would configure the channel adapter 14 to publish the event to interested process adapter in domain 1.

Thus, “receiving the data representation language schema of the service in a service advertisement of the service” is understood.

Response presented above applies to claims 18 and 45 as well.

Claims 27 and 28:

Appellant’s argument:

“Regarding claim 27, Bass fails to anticipate a service process configured to generate a message in a data representation language.”

“Bass further fails to anticipate wherein the message includes a data representation language representation of the event generated by the service process.”

“Bass also does not anticipate wherein each of the one or more event message gate units is operable to distribute the data representation language representation of the event, as asserted by the Examiner.”

Examiner’s response:

Examiner would like to reveal Bass’s teachings in steps:

Step 1:

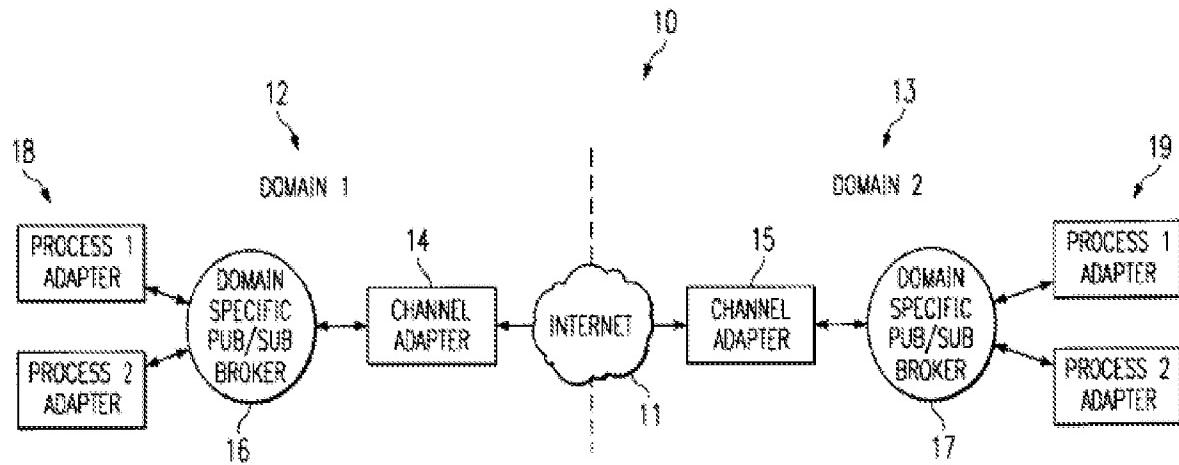


FIG. 1

Bass environment is a set up for it's teachings application at col. 3, line 5-14, "FIG. 1 depicts a high level view 10 of two domains, domain 112 and domain 213. The domains may be two separate corporations acting as business partners. Each domain represents a separate and distinct network of computers within the corporations Intranet. The two domains communicate via the Internet 11. Each domain includes an inventive channel adapter 14, 15, and a domain specific publication/subscription (PUB/SUB) service 16, 17. PUB/SUB service is the message broker service used within the domain."

Thus, communication between distinct networks (distributed computing environment), is via Internet is clearly understood.

Step 2:

Bass teaches at col. 3, line 22-42, "The inventive channel adapters 14 and 15 permit dissimilar systems to communicate with each other. The adapters acts as a proxy or an agent for communications and data exchanges between the business applications. Each adapter interfaces with its respective broker and the Internet. Note that although the Internet is depicted and described herein, other networks, such as a WAN or LAN, may be used in place of the Internet. Thus, the broker views the adapter as another process adapter within the domain to which events are published or subscribed. Furthermore, the channel adapter is configured by the domain administrator in the same manner as a process adapter in that the adapter is represented on an administrator screen as an icon which can be set to specify which events couple which processes (including the adapter). The adapter also converts communications and data into a format for transmission on the Internet (or other network). The format may use transmission control protocol/Internet protocol (TCP/IP), simple mail transport protocol (SMTP), File Transfer Protocol (FTP), or whatever protocol is useable by the connecting network."

Thus, **SMTP is used** for transmission of formatted **data** is clearly understood.

Step 3:

Bass continues teaching at col. 3, line 42-50, "Thus, the flow of information for FIG. 1 is as follows. An event originating in a process adapter 18 in domain 112 is delivered to the subscribing channel adapter 14. This event would be transformed into (for example) an e-mail via SMTP, and mailed from the channel adapter

through the Internet to domain 2 13. The e-mail is received by the channel adapter 15 and re-transformed back into the event. The channel adapter then delivers the event to any subscribing process adapters within the domain.”

Thus, **SMTP is used** for transmission of formatted **data which is “event” transformed into an e-mail** is clearly understood.

Thus, a service process is configured to generate an event wherein “a message including the data in a data representation language” is sent, which is “the data representation language representation of the event”, taught by Bass.

Further, Bass teaches that the E-mail containing data which is an event is delivered to a specific E-mail Address, as indicated in col. 4, line 3. Thus, the transformed email contains the event data and e-mail address.

Also in col. 7, line 41-45 in reference to Fig.3, Bass also teaches “Another thread is the outbound transmitting thread 305 that converts events within the outbound queue into Internet format messages, and then sends these messages to an SMTP server (or other server types) for transmission over the Internet (or other network types).”

Thus, the event converted into email contains the event data and e-mail address into the internet format messages. Thus, the event email transmitted over the internet is in a data representation language format (internet format displays “emails” on the computer screen.)

And thus, Bass anticipates “a service process is configured to generate an event” wherein “a message including the data in a data representation language” is the email, which is “the data representation language representation of the event”, is delivered to any subscribing process adapters through a configured broker. (one or more event message gate units operable to distribute the event).

Claim 29:

Appellant's argument:

“Regarding claim 29, Bass fails to anticipate a service process configured to provide a data representation language schema defining a message interface for a set of events generated by the service and also fails to teach wherein one or more event message gate units are generated according to the data representation language schema.”

Examiner's response:

Bass teaches at col. 3, line 42-50, “Thus, the flow of information for FIG. 1 is as follows. An event originating in a process adapter 18 in domain 1 12 is delivered to the subscribing channel adapter 14. This event would be transformed into (for example) an e-mail via SMTP, and mailed from the channel adapter through the Internet to domain 2 13. The e-mail is received by the channel adapter 15 and re-transformed back into the event. The channel adapter then delivers the event to any subscribing process adapters within the domain.”

Thus, “channel adapters are end points for receiving as well as delivering events between two domains” is clearly understood.

Bass teaches at col. 3, line 59-62, "Each channel adapter includes a protocol interface 22, 23. This interface comprises the network specific protocols that enable the adapter to couple with the Internet 11. The protocol interface facilitates the conversion of the events into a network transportable format, e.g. e-mail, and vice versa. Each channel adapter also includes a configuration interface 28."

Thus, the channel adapters include interfaces protocol interfaces facilitating the conversion of the events into a network transportable format, e.g. e-mail, and vice versa (wherein data representation language schema defines a message interface) is clearly understood.

Now, Bass goes on teaching at col.2, lines 4-15, "Each channel adapter is initialized with a set of events it will export to its peer at the other domain. The two channel adapters handshake with these sets of events. Process adapters within each domain can then publish events of the exported type and expect the event to be carried to the other domain via the network protocol. Likewise, a process adapter can subscribe to an event type from a channel adapter that is listed on an event type list from its peer channel adapter. When an event is received via the channel adapter and re-published into the domain, the subscribing process adapter will receive the event."(generating an event message endpoint for the client platform according to the data representation language schema, wherein said receiving a message and said sending the data representation language representation of the event to one or more processes are performed by the event message endpoint.)

Thereby the reference teaches that each channel adapter is initialized with a set of events it will export to its peer at the other domain and they handshake with sets of events even though they are in a completely different domains (wherein said data representation language schema defines a message interface for a set of events generated by the service).

Bass teaches at col. 3, line 22-42, “The inventive channel adapters 14 and 15 permit dissimilar systems to communicate with each other. The adapters acts as a proxy or an agent for communications and data exchanges between the business applications. Each adapter interfaces with its respective broker and the Internet. Note that although the Internet is depicted and described herein, other networks, such as a WAN or LAN, may be used in place of the Internet. Thus, the broker views the adapter as another process adapter within the domain to which events are published or subscribed. Furthermore, the channel adapter is configured by the domain administrator in the same manner as a process adapter in that the adapter is represented on an administrator screen as an icon which can be set to specify which events couple which processes (including the adapter). The adapter also converts communications and data into a format for transmission on the Internet (or other network). The format may use transmission control protocol/Internet protocol (TCP/IP), simple mail transport protocol (SMTP), File Transfer Protocol (FTP), or whatever protocol is useable by the connecting network.”

Also, Bass teaches at col. 4, line 64-66,” The administrator would configure the broker 17 to list channel adapter 15 as a subscriber to the event.”

Thus, Bass anticipates “wherein one or more event message gate units are generated according to the data representation language schema.”

Claim 30:

Appellant's argument:

“Regarding claim 30, Bass fails to anticipate wherein the data representation language schema defines a set of messages that the service may send to the event message gate units.”

Examiner's response:

Bass teaches at col. 4, line 64-66, “The administrator would configure the broker 17 to list channel adapter 15 as a subscriber to the event.”

Thus, Bass anticipates “the data representation language schema defines a set of messages that the service may send to the event message gate units.”

Claim 31:

Appellant's argument:

“Regarding claim 31, Bass does not teach a service process configured to provide the data representation language schema in a service advertisement.”

Examiner's response:

As indicated above for claim 2, Bass teaches at col. 3, line 42-50, “Thus, the flow of information for FIG. 1 is as follows. An event originating in a process adapter 18 in domain 1 12 is delivered to the subscribing channel adapter 14. This event would be transformed into (for example) an e-mail via SMTP, and mailed from the channel adapter through the Internet to domain 2 13. The e-mail is received by the channel

adapter 15 and re-transformed back into the event. The channel adapter then delivers the event to any subscribing process adapters within the domain."

Thus, "channel adapters are end points for receiving as well as delivering events between two domains" is clearly understood.

Bass teaches at col. 3, line 59-62, "Each channel adapter includes a protocol interface 22, 23. This interface comprises the network specific protocols that enable the adapter to couple with the Internet 11. The protocol interface facilitates the conversion of the events into a network transportable format, e.g. e-mail, and vice versa. Each channel adapter also includes a configuration interface 28."

Thus the channel adapters include interfaces protocol interfaces facilitating the conversion of the events into a network transportable format, e.g. e-mail, and vice versa (wherein data representation language schema defines a message interface) is clearly understood.

Now, Bass goes on teaching at col.2, lines 4-15, "Each channel adapter is initialized with a set of events it will export to its peer at the other domain. The two channel adapters handshake with these sets of events. Process adapters within each domain can then publish events of the exported type and expect the event to be carried to the other domain via the network protocol. Likewise, a process adapter can subscribe to an event type from a channel adapter that is listed on an event type list from its peer channel adapter. When an event is received via the channel adapter and re-published into the domain, the subscribing process adapter will receive the event."(generating an event message endpoint for the client platform according to the data representation

language schema, wherein said receiving a message and said sending the data representation language representation of the event to one or more processes are performed by the event message endpoint.)

Thereby the reference teaches that each channel adapter is initialized with a set of events it will export to its peer at the other domain and they handshake with sets of events even though they are in a completely different domains (receiving a data representation language schema on the client platform, wherein said data representation language schema defines a message interface for a set of events generated by the service).

Now that Bass has taught that the “data representation language schema defines a message interface for a set of events generated by the service” , Bass teaches at col.4, line 57 through col.5, line 15, “Thus, prior to transfer of events between the domains, the respective process and channel adapters of the domains must be configured to send and receive the different events. The mechanism for configuring the adapters is as follows. The configuration begins when an adapter signals the administrator that it has a new event to publish to interested processes. For example, assume process adapter 119 of domain 213 is publishing the new event. The administrator would configure the broker 17 to list channel adapter 15 as a subscriber to the event. Thus, adapter 15 would receive the event from the process, when the process adapter publishes the event. The administrator, via configuration interface 29, would configure the channel adapter to send the event over the Internet to channel adapter 14 of domain 1. The channel adapter 15 would signal the channel adapter 14

that a new event is going to be received by channel adapter 14, and sends the updated export list to the other channel adapter. The channel adapter of domain 2 becomes capable of exporting the event. The administrator of domain 1 would configure the channel adapter 14 to publish the event to interested process adapter in domain 1.

Thus, “a service process configured to provide the data representation language schema in a service advertisement.”” is clearly understood.

Claim 32:

Appellant’s argument:

“Regarding claim 32, Bass fails to teach the event message endpoint subscribing to one or more of the set of events generated by the service, wherein the service is configured to send messages including data representation language representations of an event to subscribers to the event when the event is generated.”

Examiner’s response:

Bass goes on teaching at col.2, lines 4-15, "Each channel adapter is initialized with a set of events it will export to its peer at the other domain. The two channel adapters handshake with these sets of events. Process adapters within each domain can then publish events of the exported type and expect the event to be carried to the other domain via the network protocol. Likewise, a process adapter can subscribe to an event type from a channel adapter that is listed on an event type list from its peer channel adapter. When an event is received via the channel adapter and re-published into the domain, the subscribing process adapter will receive the event."

Thus, “channel adapters are end points for receiving as well as delivering events between two domains” is clearly understood.

And “a process adapter can subscribe to an event type from a channel adapter that is listed on an event type list from its peer channel adapter” is clearly understood.

And, thereby the event message endpoint subscribing to one or more of the set of events generated by the service, wherein the service is configured to send messages including data representation language representations of an event, as explained in claim 1, to subscribers to the event when the event is generated.” is taught by Bass.

Claim 33:

Appellant’s argument:

“In regards to claim 33, Bass fails to disclose wherein the service process is further configured to attach an authentication credential for the service to the data representation language message, where the authentication credential is configured for use in authenticating the data representation language message as being from the service process.”

Examiner’s response:

Bass teaches at col. 3, line 64 through col. 4, line 3, “Each channel adapter also includes a configuration interface 28. This interface allows the channel adapter to be configured by an administrator. The administrator (not shown) could select the protocols used for the interfaces and specify what events or types of events would flow through the channel adapter and configures protocol specific parameters, e.g. e-mail addresses.”

Thus, “service includes an authentication credential for the service.” is clearly understood.

Second Ground of Rejection:

Claim 12, 25 and 47:

Appellant’s argument:

“Contrary to the Examiner's assertion, Bass in view of Meltzer fails to teach or suggest a message in a data representation language including a data representation language representation of an event, where the event is a JAVA event.”

“The Examiner admits that Bass does not disclose the use of XML or JAVA events.”

Examiner’s response:

Appellant is absolutely right that “Bass does not disclose the use of XML or JAVA events.” But, it does not mean that Bass does not teach “a message in a data representation language including a data representation language representation of an event”, which it does as previously stated.

Bass’s event communication is, as disclosed by Bass at col. 3, line 22-26 , “The inventive channel adapters 14 and 15 permit dissimilar systems to communicate with each other. The adapters acts as a proxy or an agent for communications and data exchanges between the business applications. Each adapter interfaces with its respective broker and the Internet. “, between business applications.

Meltzer teaches "By translating the elements of the XML document into JAVA events or other programming structures that are suitable for use by the transaction

processing front end of the respective nodes enables rich functionality at nodes utilizing the documents being traded." (col.14, lines 25-32).

Thus Meltzer provides a concept that is of paramount importance and that is translating the event in a data representation language into a JAVA event, which makes obvious to one having ordinary skill in the art at the time of invention was made to incorporate the teachings of Meltzer into the teachings of Bass such that the transaction process front end (channel adapters are configured to communicate between business applications) is able to operate in a publish and subscribe architecture that enables the addition of new listener programs without the knowledge of or impact on other listening programs in the system as taught by the reference.

Claim 13, 26 and 48:

Appellant's argument:

"Thus, Meltzer does not teach or suggest sending messages in a XML, nor does Meltzer teach or suggest including an XML representation of an event in such a message. In fact, Meltzer teach away from including XML representations of events."

Examiner's response:

Bass's event communication is, as disclosed by Bass at col. 3, line 22-26, "The inventive channel adapters 14 and 15 permit dissimilar systems to communicate with each other. The adapters acts as a proxy or an agent for communications and data exchanges between the business applications. Each adapter interfaces with its respective broker and the Internet. ", between business applications.

Meltzer teaches "By translating the elements of the XML document into JAVA events or other programming structures that are suitable for use by the transaction processing front end of the respective nodes enables rich functionality at nodes utilizing the documents being traded." (col.14, lines 25-32).

Meltzer also discloses in Abstract, "The self defining electronic documents, such as XML based documents, can be easily understood amongst the partners." which makes obvious to one having ordinary skill in the art at the time of invention was made to incorporate the teachings of Meltzer into the teachings of Bass such that the transaction process front end (channel adapters are configured to communicate between business applications) is able to operate in a publish and subscribe architecture that enables the addition of new listener programs without the knowledge of or impact on other listening programs in the system as taught by the reference.

Claim 34:

Appellant's argument:

"Contrary to the Examiner's assertion, Bass in view of Meltzer fails to teach or suggest a message in a data representation language including a data representation language representation of an event, where the event is a JAVA event. The Examiner admits that Bass does not disclose the use of XML or JAVA events."

Examiner's response:

Appellant is absolutely right that "Bass does not disclose the use of XML or JAVA events." But, it does not mean that Bass does not teach "a message in a data

representation language including a data representation language representation of an event", which it does, as previously stated.

Bass's event communication is, as disclosed by Bass at col. 3, line 22-26, "The inventive channel adapters 14 and 15 permit dissimilar systems to communicate with each other. The adapters acts as a proxy or an agent for communications and data exchanges between the business applications. Each adapter interfaces with its respective broker and the Internet. ", between business applications.

Meltzer teaches "By translating the elements of the XML document into JAVA events or other programming structures that are suitable for use by the transaction processing front end of the respective nodes enables rich functionality at nodes utilizing the documents being traded." (col.14, lines 25-32).

Thus Meltzer provides a concept that is of paramount importance and that is translating the event in a data representation language into a JAVA event, which makes obvious to one having ordinary skill in the art at the time of invention was made to incorporate the teachings of Meltzer into the teachings of Bass such that the transaction process front end (channel adapters are configured to communicate between business applications) is able to operate in a publish and subscribe architecture that enables the addition of new listener programs without the knowledge of or impact on other listening programs in the system as taught by the reference.

Claim 35:

Appellant's argument:

"Nowhere does Meltzer teach or suggest including an data representation language representation of an event in a message in a data representation language, where the data representation language is XML."

Examiner's response:

Bass's event communication is, as disclosed by Bass at col. 3, line 22-26, "The inventive channel adapters 14 and 15 permit dissimilar systems to communicate with each other. The adapters acts as a proxy or an agent for communications and data exchanges between the business applications. Each adapter interfaces with its respective broker and the Internet. ", between business applications.

Meltzer teaches "By translating the elements of the XML document into JAVA events or other programming structures that are suitable for use by the transaction processing front end of the respective nodes enables rich functionality at nodes utilizing the documents being traded." (col.14, lines 25-32).

Meltzer also discloses in Abstract, "The self defining electronic documents, such as XML based documents, can be easily understood amongst the partners." which makes obvious to one having ordinary skill in the art at the time of invention was made to incorporate the teachings of Meltzer into the teachings of Bass such that the transaction process front end (channel adapters are configured to communicate between business applications) is able to operate in a publish and subscribe architecture that enables the addition of new listener programs without the knowledge of or impact on other listening programs in the system as taught by the reference.

(11) Related Proceeding(s) Appendix

No decision rendered by a court or the Board is identified by the examiner in the Related Appeals and Interferences section of this examiner's answer.

For the above reasons, it is believed that the rejections should be sustained.

Respectfully submitted,

/Ashok B. Patel/

Ashok B. Patel
Examiner
Art Unit 2154

Conferees:

/Nathan J. Flynn/

Supervisory Patent Examiner, Art Unit 2154

/John Follansbee/

Supervisory Patent Examiner, Art Unit 2151

Meyertons, Hood, Kivlin, Kowert & Goetzel, P.C.
P.O. Box 398
Austin, TX 78767-0398
(512) 853-8850

Application/Control Number: 09/692,765
Art Unit: 2100

Page 57